AMENDMENTS TO THE SPECIFICATION:

Please amend the paragraph beginning at page 8, line 10, as follows:

Figure 6 is an illustrative view showing an example state in which two backup areas of a flash memory as illustrated by Figure 2 are provided, wherein [[when]] one backup area is becomes unwritable[[,]] and the other backup area is prohibited from being written;

Please amend the paragraph beginning at page 8, line 16, as follows:

Figure 8 is an illustrative view showing a state in which two backup areas of a flash memory as illustrated by Figure 2 are provided and a writing prohibiting flag area is further provided, wherein [[when]] one backup area becomes is unwritable[[,]] and the other backup area is prohibited from being written;

Please amend the paragraph beginning at page 8, line 24, as follows:

Figure 11 is a flowchart showing an example backup area selection process of performed by the backup areas of the CPU shown in Figure 2;

Please amend the paragraph beginning at page 8, line 26, as follows:

Figure 12 is a flowchart showing an example renewal data renewal process of performed by historical data of the CPU shown in Figure 2;

Please amend the paragraph beginning at page 9, line 1, as follows:

1288941

Figure 13 is an illustrative view showing [[one]] another non-limiting illustrative example of a memory map of a flash memory provided in a game cartridge ~~loaded in another game machine~~;

Please amend the paragraph beginning at page 9, line 4, as follows:

Figure 14 is a flowchart showing a part of a backup process performed by the CPU for a memory map as shown in Figure 13;

Please amend the paragraph beginning at page 9, line 6, as follows:

Figure 15 is a flowchart showing another part of the backup process performed by the CPU for a memory map as shown in Figure 13; and

Please amend the paragraph beginning at page 21, line 19, as follows:

It is noted that the backup area is first selected from n=1 in order until the game data is stored in all the backup areas 58n in this ~~embodiment~~ example implementation; however, if only the writing is not performed on the same backup area, the backup area may be selected by an arbitrary method.

Please amend the paragraph beginning at page 21, line 22, as follows:

The condition for area selection in this ~~embodiment~~ implementation is as shown in ~~Figure 2~~ Table 2 (below), and the condition for area selection to be utilized is, as the above-described ~~embodiment~~ implementation, determined in advance in correspondence to the historical data of

the last game data. For example, if the historical data of the last game data is "1", the condition

for area selection utilized is 2N > 2N-1 > ... >N+2 > N+1.

Please amend the paragraph beginning at page 12, line 24, as follows:

Furthermore, the condition for area selection is represented by four inequalities of 2>1,

3>2, 4>3, 1>4. These numerals 1 to 4 are numerical values indicated by the historical data and

mean that the game data including the historical data of a numeral on the right side is older than

the game data including the historical data of a numeral on the left side (written to the flash

memory 58 earlier). Such the condition for area selection is determined (defined) by inequalities

(conditions) utilized in correspondence to the historical data of the last game data, and

specifically represented by a table 1

Table 1

Historical Data of Last Game Data	Condition for Area Selection
1	4>3 (3 is older than 4)
2	1>4 (4 is older than 1)
3	2>1 (1 is older than 2)
4	3>2 (2 is older than 3)

Please amend the paragraph beginning at page 14, line 12, as follows:

Specifically, as shown in Figure 6, in a case the game data including the historical data (1)

is written to the first backup area 58a and the game data including the historical data (4) is

written to the second backup area 58b, as the write-objective area for the last game data

including the historical data (2), the second backup area 58b is selected according to the

condition of the area selection [[(4 > 1)]] (1 > 4) . Then, a writing process of the last game data

- 5 -

is executed. In the writing process of the game data, it is determined whether or not the game data is normally written by a checksum, and in a case the normal writing is not performed, the writing process is repeated by plural number of times (e.g., three times). As a consequence of the writing plural times, if failing in the writing of the game data, it is determined to be unwritable. Then, the writing process of the game data is forcedly terminated so as to prohibit the writing of the last game data to another area (first backup area 58a in Figure 6).